

## Manuel d'utilisation TP Log Apache

### I. Compilation et exécution

Nous avons créé un makefile pour faciliter la compilation. Il se trouve dans le dossier **bin**, avec tous les fichiers ".h" et ".cpp".

Pour l'exécution, nous pouvons utiliser la syntaxe suivante : `./analog [-options] <fichier_test>.log`

### II. Tests

Nous avons réalisé en tout 12 tests, dont les fichiers ".log" sont dans le dossier **LogTest** et le framework est dans le dossier **Test**.

#### Description des tests :

1. Exécution sans option
2. Exécution sans option mais avec un fichier contenant moins de 10 logs différents
3. Exécution d'un fichier ".log" vide
4. Exécution sans fichier ".log" spécifié
5. Exécution avec fichier ".log" inexistant
6. Exécution avec l'option -e
7. Exécution avec l'option -e mais avec un fichier ne contenant aucune image
8. Exécution avec l'option -e avec un fichier ne contenant que des images
9. Exécution avec l'option -t permet de sélectionner les fichiers avec une tranche horaire
10. Exécution avec une option inexistante
11. Exécution avec -e et -t en même temps sur un fichier log qui contient des fichiers de format image
12. Exécution avec l'option -g pour produire un fichier au format GraphViz

### III. Graphe

La création de "graphe dessiné" passe par la création de fichier .dot, crée avec l'option -g. Pour les visualiser il faut utiliser un logiciel adéquat comme GraphViz.

Par exemple : `./analog -g monGraphe.dot log1.log` , qui créera un fichier monGraphe.dot avec les informations contenues dans le fichier log1.log

### IV. Comment lancer les tests

1. `./test.sh Test1`
2. `./test.sh Test2`
3. `./test.sh Test3`
4. `./test.sh Test4`
5. `./test.sh Test5`
6. `./test.sh Test6`
7. `./test.sh Test7`
8. `./test.sh Test8`
9. `./test.sh Test9`
10. `./test.sh Test10`
11. `./test.sh Test11`
12. `./test.sh Test12`